МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ТЕХНОЛОГІЇ ШТУЧНОГО ІНТЕЛЕКТУ-2

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання практикуму для студентів спеціальності

„Автоматизоване управління технологічними процесами”

Київ

НТУУ “КПІ”

2014

Технології штучного інтелекту-2. Комп'ютерні технології інтелектуального аналізу даних: Метод. вказівки до викон. практикуму для студ. спец. „Автоматизоване управління технологічними процесами” / Уклад.: Д.О. Ковалюк. – К. : НТУУ ”КПІ“, 2014. – 26с.

Гриф надано Вченою радою ІХФ
(Протокол № 2 від 24 лютого 2014 р.)

Навчальне видання

ТЕХНОЛОГІЇ ШТУЧНОГО ІНТЕЛЕКТУ-2

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Методичні вказівки до виконання практикуму для студентів спеціальності

„Автоматизоване управління технологічними процесами”

Методичні вказівки призначено для виконання практичних занять з дисципліни «Технології штучного інтелекту».

Укладач: Ковалюк Дмитро Олександрович,
канд. техн. наук, доц.

Відповідальний редактор А.І. Жученко, докт. техн. наук, проф.

Рецензент А.Р. Степанюк, канд. техн. наук, доц.

Авторська редакція

ЗМІСТ

Вступ…………………………………………………………………….. 4

Мета і завдання практичних занять…..…………………………….. 5

Практичне заняття 1

Розв’язання задачі класифікації методом Naive Bayes ......…..……… 6

Практичне заняття 2

Розв’язання задачі класифікації методом найближчого сусіда……... 12

Практичне заняття 3

Розв’язання задачі класифікації методом опорних векторів…..……... 17

Практичне заняття 4

Розв’язання задачі прогнозування на основі нейронних мереж …….. 21

ВСТУП

Підґрунтям нового розвитку задачі аналізу даних є стрімкий науково-технічний прогрес і збільшення потужностей обчислювальної техніки. Це дало змогу зберігати великі об’єми даних, що характеризують певну предметну область. Обробка таких даних є невід’ємною частиною побудови та функціонування інтелектуальних систем керування, і використовується в них для створення або уточнення математичних моделей процесу, прийняття рішень.

«Класичні» методи аналізу даних почали активно розроблятися з середини XX століття. До них зокрема відносяться методи математичної статистики (кореляційний, регресійний, факторний аналіз), методи теорії планування експерименту. Дані групи методів знайшли широке практичне використання, що підтвердило їх доцільність та значимість.

Проте, із збільшенням об’єму наявної інформації постала задача не лише її статистичної обробки, але й виявлення прихованих закономірностей та залежностей в даних. Цей напрямок зумовив розвиток нової технології – DataMining.

Реалізації методів DataMining можлива у вигляді комп’ютерних технологій, представлених спеціальними програмами або власним програмним забезпеченням алгоритмів аналізу даних.

МЕТА І ЗАВДАННЯ ПРАКТИЧНИХ ЗАНЯТЬ

Метою практичних занять є формування у студентів компетенцій щодо одержання знань та вмінь для інтелектуального аналізу даних.

В результаті студенти закріплюють наступні знання:

- задачі інтелектуального аналізу даних;
- типи моделей представлення даних;
- алгоритми розв'язання задач класифікації та прогнозування;
- технології та засоби для реалізації алгоритмів інтелектуального аналізу даних;

Після практичних занять студенти повинні вміти:

- вибирати найбільш доцільні методи обробки даних;
- створювати математичні моделі процесів з використанням інтелектуальних методів аналізу даних – нейронних мереж, дерев рішень;
- аналізувати показники адекватності розроблених моделей;
- використовувати наявні та створювати власні комп'ютерні технології інтелектуального аналізу даних.

ПРАКТИЧНЕ ЗАНЯТТЯ №1

РОЗВ'ЯЗАННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ МЕТОДОМ NAIVE BAYES

Мета роботи – ознайомитися з Байесовим методом класифікації, побудувати класифікатор для прогнозування утворення пробки в процесі екструзії, виконати класифікацію нового екземпляра.

Постановка задачі

Екструзія являє собою безперервний технологічний процес, який полягає в продавлюванні матеріалу, що володіє високою в'язкістю в рідкому стані, через формуючий інструмент (екструзійну головку) з метою отримання виробу з поперечним перерізом потрібної форми.

У промисловості переробки полімерів методом екструзії виготовляють труби, плівки, листи, профільні вироби, наносять ізоляцію на кабелі тощо.

В процесі екструзії полімер спресовується та ущільнюється, що може викликати утворення твердої пробки у просторі між витками шнеку і внутрішньою поверхнею корпусу (далі – міжвитковий простір). Це в свою чергу призводить до аварійної ситуації через неможливість обертання шнека.

Фактори, що визначають появу пробки, наступні:

1. кількість обертів екструдера (об/хв);
2. витрата сировини (кг/год);
3. щільність та гранулометричний склад (діаметр гранул - мм);
4. температура в екструдері (^{0}C).

Статистична інформація, наявна в даному процесі, представлена в таблиці 1.1.

Таблиця 1.1.

К-ть обертів	**Витрата**	**Щільність**	**Розмір гранул**	**Температура**	**Пробка**
120	30	низька	3	165	ні
110	35	низька	2	150	ні
130	40	висока	7	155	так
120	35	низька	5	165	ні
130	25	висока	2	180	ні
90	60	висока	6	150	так
100	50	низька	5	150	так
140	30	висока	3	200	ні
110	40	висока	3	160	ні
120	50	висока	2	155	ні
130	35	низька	4	165	ні
120	40	висока	5	150	так
115	35	низька	4	155	ні
135	40	висока	7	150	так
120	35	низька	4	170	ні
115	25	висока	3	140	ні
100	60	висока	3	180	так
100	50	висока	2	175	ні
140	30	низька	3	200	ні
110	50	висока	4	165	так
120	25	низька	2	180	ні
130	30	низька	3	200	ні

Необхідно виконати задачу класифікації утворення пробки методом Naive Bayes за таких умов:

К-ть обертів	Витрата	Щільність	Розмір гранул	Температура	Пробка
130	40	низька	5	165	?

Алгоритм Байеса

Метод Байесової класифікації є статистичним методом і дозволяє розрахувати ймовірність належності об'єкта до заданого класу. Оснований на теоремі Байеса.

Нехай подія A може наступити за умови появи однієї з несумісних подій B_1, B_2, …, B_n, які утворюють повну групу. Нехай відомі ймовірності цих подій та умовні ймовірності $P_{B_1}(A)$, $P_{B_2}(A)$, …,$P_{B_n}(A)$ події A.

Ймовірність події A, яка може настати лише за умови появи однієї з несумісних подій B_1, B_2, …, B_n, що утворюють повну групу, дорівнює сумі добутків ймовірностей кожної з цих подій на відповідну умовну ймовірність події A.

$$P(A) = P(B_1) \cdot P_{B_1}(A) + P(B_2) \cdot P_{B_2}(A) + ... + P(B_n) \cdot P_{B_n}(A)$$

Оскільки невідомо, яка з подій B_1, B_2, …, B_n наступить, їх називають гіпотезами. Припустимо, що було здійснено випробування і наступила подія A. Тоді, використовуючи цю інформацію можна перерахувати умовні ймовірності подій $P_{B_1}(A)$, $P_{B_2}(A)$, …,$P_{B_n}(A)$ за формулою

$$P_{B_i}(A) = \frac{P(B_i) \cdot P_{B_i}(A)}{P(B_1) \cdot P_{B_1}(A) + P(B_2) \cdot P_{B_2}(A) + ... + P(B_n) \cdot P_{B_n}(A)}$$

Формула Байеса дозволяє переоцінити ймовірність гіпотез, після появи події A, коли відомо, яка з подій B_1, B_2, …, B_n призвела до цього.

Побудова класифікатора

Дано об'єкт $\mathbf{X} = \{x_1, x_2, ...x_n\}$, клас якого невідомий. В даній задачі об'єкт $\mathbf{X}$ - це стан екструдера, а атрибути $\{x_1, x_2, ...x_n\}$ – це параметри процесу.

Кожен об'єкт може бути віднесений до одного з двох класів:

- C_1 - пробка утвориться
- C_2 - пробка не утвориться

Відповідно до цього висувається дві гіпотези:

1. Утворення пробки в екструдері (гіпотеза H1)
2. Пробка в екструдері не утвориться (гіпотеза H2)

Ідея Байесового класифікатора полягає в тому, щоб розрахувати ймовірності гіпотез $P(H \mid \mathbf{X})$. Розраховані ймовірності $P(H_i \mid \mathbf{X})$, показують з якою ймовірністю об'єкт відноситься до i-го класу. Для нашої задачі:

$P(H_1 \mid \mathbf{X})$ – ймовірність, що об'єкт відноситься до класу C_1

$P(H_2 \mid \mathbf{X})$ – ймовірність, що об'єкт відноситься до класу C_2

Ці ймовірності розраховують за формулою Байеса:

$$P(H_i \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H_i) \cdot P(H_i)}{P(\mathbf{X})}$$

де

$P(H_i \mid \mathbf{X})$ – ймовірність гіпотези, що об'єкт відноситься до i-го класу;

$P(H_i)$ – апріорна ймовірність появи кожного з класів (не залежить від $\mathbf{X}$);

$P(\mathbf{X})$ – апріорна ймовірність атрибутів $\mathbf{X}$;

$P(\mathbf{X} \mid H_i)$ – це апостеріорна ймовірність, що об'єкт характеризується атрибутами $\mathbf{X} = \{x_1, x_2, \dots x_n\}$, за умови, що він належить до класу H_i

Апріорні ймовірності появи кожного з класів:

P(пробка = так) =7/22

P(пробка = ні) =15/22

Апостеріорні ймовірності

P (к-ть обертів = 115 об/хв. | пробка = так) = 1/7

P (к-ть обертів = 115 об/хв. | пробка = ні) = 3/15

P (витрата = 40 кг/год | пробка = так) = 2/7

P (витрата = 40 кг/год | пробка = ні) = 1/15

P (щільність = «низька» | пробка = так) = 1/7

P (щільність = «низька» | пробка = ні) = 9/15

P (розмір гранул = 5 мм | пробка = так) = 2/7

P (розмір гранул = 5 мм | пробка = ні) = 1/15

P (температура = 165^0C | пробка = так) = 1/7

P (температура = 165^0C | пробка = ні) = 3/15

Ймовірності належності об'єкта до кожного з класів:

P(пробка = так|X) = P(к-ть обертів = 115 об/хв|пробка = так) × P (витрата = 40 кг/год|пробка = так) × P(щільність = «низька»|пробка = так) × P(розмір гранул= 5мм|пробка = так) × P(температура = 165^0C|пробка = так) × P(пробка = так)

P(пробка = ні|X) = P(к-ть обертів = 115 об/хв|пробка = ні) × P(витрата = 40 кг/год|пробка = ні) × P(щільність = «низька»|пробка = ні)× P(розмір гранул = 5 мм|пробка = ні) × P(температура = 165^0C|пробка = ні) × P(пробка = ні)

Підставивши наступні ймовірності отримаємо :

P (пробка = так | X) = $\frac{1}{7}\cdot\frac{2}{7}\cdot\frac{1}{7}\cdot\frac{2}{7}\cdot\frac{1}{7}\cdot\frac{7}{22} = 7.573\times 10^{-5}$

P (пробка = ні | X) = $\frac{3}{15}\cdot\frac{1}{15}\cdot\frac{9}{15}\cdot\frac{1}{15}\cdot\frac{3}{15}\cdot\frac{15}{22} = 7.273\times 10^{-5}$

В даному випадку можна стверджувати, що при вказаних умовах пробка утвориться / не утвориться з ймовірністю:

$$P'(\text{пробка} = \text{так} \mid X) = \frac{7.573 \times 10^{-5}}{7.573 \times 10^{-5} + 7.273 \times 10^{-5}} = 0.51$$

$$P'(\text{пробка} = \text{ні} \mid X) = \frac{7.273 \times 10^{-5}}{7.573 \times 10^{-5} + 7.273 \times 10^{-5}} = 0.49$$

Література

1. Барсегян А.А., Куприянов М.С., Степаненко В.В Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004.–336с.
2. Гайдышев И. Г. Анализ и обработка данных: специальный справочник — СПб: Питер, 2001. — 752 с.

Контрольні запитання

1. Апріорні та апостеріорні ймовірності.
2. Формула повної ймовірності та формула Байеса.
3. Складові формули Байеса в задачі класифікації.
4. «Наївний» Байесовий класифікатор, припущення, реалізація.
5. Нормалізація ймовірностей гіпотез.

ПРАКТИЧНЕ ЗАНЯТТЯ №2

РОЗВ’ЯЗАННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ІРИСІВ МЕТОДОМ НАЙБЛИЖЧОГО СУСІДА

Мета роботи – ознайомитися з методом найближчого сусіда, побудувати класифікатор для статистичного набору даних, виконати класифікацію нового екземпляра.

Постановка задачі

Є набір даних, зібраних Р. Фішером, про 150 квіток трьох класів: IrisSetosa, IrisVersicolour, IrisVirginica, по 50 записів для кожного. Відома довжина і ширина чашолистка і пелюстки всіх цих ірисів.

Для простоти розглянемо тільки два вхідних атрибути об’єкту: довжину чашолистки і довжину пелюстки, а також вихідний – клас квітки. Необхідно визначити клас екземпляра з наступними значеннями довжини чашолистка і пелюстки - 6,2 і 4,9 (в сантиметрах).

Алгоритм методу найближчого сусіда

Нехай є n спостережень, кожному з яких відповідає запис в таблиці. Всі записи належать якомусь класу. Необхідно визначити клас для нового запису. На першому кроці алгоритму задається число k - кількість найближчих сусідів, вже відомих екземплярів по класу яких робиться висновок про клас нового об’єкту. Зазначимо, що якщо прийняти k=1, то алгоритм втратить узагальнюючу здатність, так як новому запису буде присвоєно клас найближчий до нього. Якщо встановити занадто велике значення, то багато локальних особливостей не буде виявлено.

На другому кроці знаходяться k-записів з мінімальною відстанню до вектора ознак нового об'єкта (пошук сусідів). Функція для розрахунку відстані повинна відповідати наступним правилам:

1. $d(x, y) \geq 0, d(x, y) = 0$ тоді і тільки тоді, коли $x = y$;
2. $d(x, y) = d(y, x)$;
3. $d(x, z) \leq d(x, y) + d(y, z)$, при умові, що точки $x,\ y,\ z$ не лежать в одній прямій.

Для впорядкованих значень атрибутів знаходиться Евклідова відстань:

$$D_E = \sqrt{\sum_{i}^{n} (x_i - y_i)^2}, \qquad (2.1)$$

Де n-кількість атрибутів.

Для інших типів змінних, які не можуть бути впорядковані, може бути застосована функція відмінності, яка задається наступним чином:

$$d(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases}, \qquad (2.2)$$

При знаходженні відстані може бути врахована значущість атрибутів, яка визначається експертом або аналітиком суб'єктивно.

На третьому кроці, коли знайдені записи, найбільш ближчі до нового, необхідно вирішити, як вони впливають на клас нового запису. Є декілька варіантів такого вибору, одним з яких є просте не виважене голосування. Ідея цього способу полягає в тому, що новому екземпляру присвоюється клас, до якого відносяться більшість з сусідів

Побудова класифікатора

Метод найближчих сусідів уже реалізований в Statistic Toolbox пакету Matlab. Для реалізації даного методу в Matlab [2] виконуються наступні кроки:

1. Завантаження вхідних даних. Завантажимо набір тестових даних, візуалізуємо його та позначимо на графіку новий екземпляр, який необхідно класифікувати:

```
load fisheriris
x = [meas(:,1) meas(:,3)];
gscatter(x(:,1), x(:,2), species, 'rgb')
xlabel('довжина чашолистка');
ylabel('довжина пелюстки');
set(legend, 'location', 'best')
hold on
newpoint = [6.2 4.9];
line(newpoint(1), newpoint(2), 'marker', 'x', 'color',
'black',  'markersize',8,'linewidth',2)
```

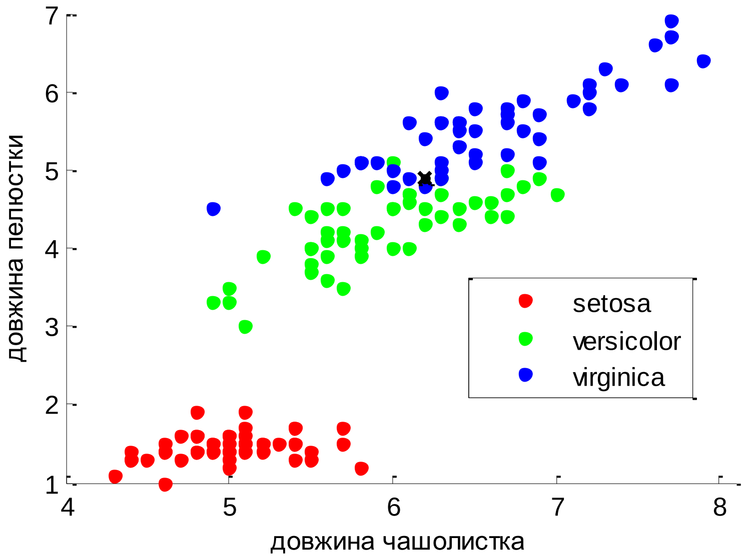

Рис.2.1. Статистичні дані та новий екземпляр

2. Визначення найближчих сусідів. Задамо значення кількості найближчих сусідів - 10, і отримаємо їх характеристики:

```
[n,d] = knnsearch(x, newpoint, 'k', 10)
class = species(n,1)
iris = [x(n,1), x(n,2)]
```

Масив [n,d] містить порядковий номер сусіда у навчальній вибірці, та відстань від сусіда до нового об’єкту:

```
n = 73 124 127 128 147 134 57 120 64 74
```

```
d = Columns 1 through 5
0.1000  0.1000  0.1000  0.1000  0.1414
Columns 6 through 10
0.2236 0.2236  0.2236  0.2236  0.2236
```

Для візуального представлення найближчих сусідів використовується код:

```
line(x(n,1), x(n,2), 'color', [.5 .5 .5], 'marker', 'o',...
    'linestyle', 'none', 'markersize',10)
```

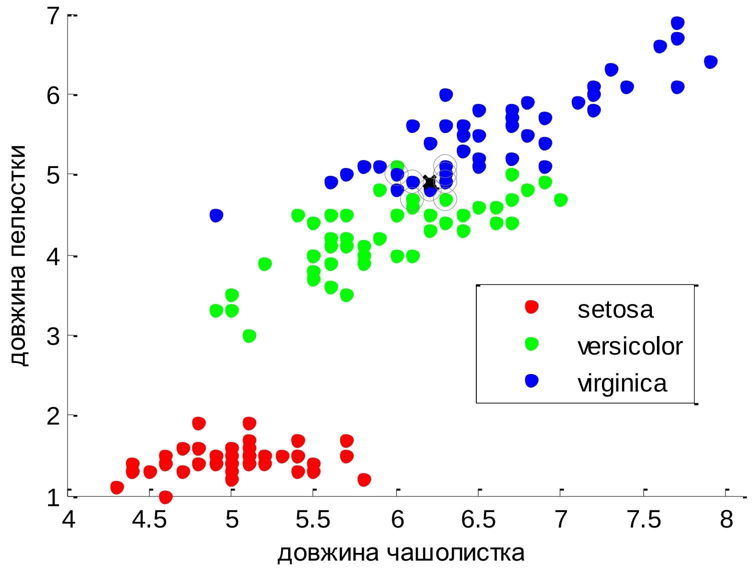

Рис.2.2. Графічна інтерпретація 10 найближчих сусідів

Або у збільшеному вигляді:

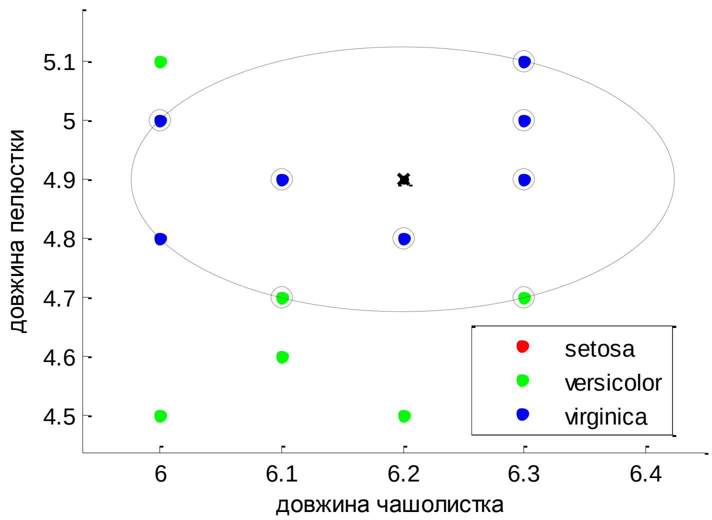

Рис.2.3. Графічна інтерпретація 10 сусідів - збільшений вигляд

На графіку представлено лише 8 точок, так як 2 з них мають однакові координати, що ілюструє наведені нижче значення:

Фактори класифікації		Клас
ans =		class =
6.3000	4.9000	'versicolor'
6.3000	4.9000	'virginica'
6.2000	4.8000	'virginica'
6.1000	4.9000	'virginica'
6.3000	5.0000	'virginica'
6.3000	5.1000	'virginica'
6.3000	4.7000	'versicolor'
6.0000	5.0000	'virginica'
6.1000	4.7000	'versicolor'
6.1000	4.7000	'versicolor'

3. Визначення класу нового об'єкту. За допомогою команди tabulate визначимо процентне співвідношення знайдених сусідів для прийняття рішення.

```
tabulate(species(n))

Value     Count    Percent
versicolor   4       40.00%
virginica    6       60.00%
```

Припустимо, що на останньому етапі алгоритму використовується просте незважене голосування експертів. Тоді, можна зробити висновок про належність об'єкту до класу Virginica.

Література

1. Барсегян А.А., Куприянов М.С., Степаненко В.В Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004.–336с.

2. Statistics Toolbox™ User's Guide

Контрольні запитання

1. Основна ідея методу найближчих сусідів.
2. Міри визначення близькості.
3. Процедури визначення класу об'єкту.

ПРАКТИЧНЕ ЗАНЯТТЯ №3

РОЗВ’ЯЗАННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ МЕТОДОМ ОПОРНИХ ВЕКТОРІВ

Мета роботи – ознайомитися з методом опорних векторів, побудувати класифікатор для статистичного набору даних, виконати класифікацію нового екземпляра.

Постановка задачі

Є набір даних, про 150 квіток трьох класів: Iris Setosa, Iris Versicolour, Iris Virginica, по 50 записів для кожного. Відома довжина і ширина чашолистка і пелюстки всіх цих ірисів.

Для простоти розглянемо тільки два вхідних атрибути об’єкту: довжина чашолистки і довжина пелюстки, а також вихідний – клас квітки. Необхідно визначити клас екземплярів з наступними значеннями довжини чашолистка і пелюстки в сантиметрах: (5.2; 1,9) та (6.2; 4,9).

Алгоритм методу

Метод опорних векторів (SVM - support vector machines) - належить до сімейства лінійних класифікаторів. Основна ідея - переведення вихідних векторів у простір більш високої розмірності та пошук розділяючої гіперплощини з максимальним зазором в цьому просторі. Дві паралельні гіперплощини будуються по обидва боки гіперплощини, що розділяє наші класи. Розділяючою гіперплощиною буде гіперплощина, що максимізує відстань до двох паралельних гіперплощин.

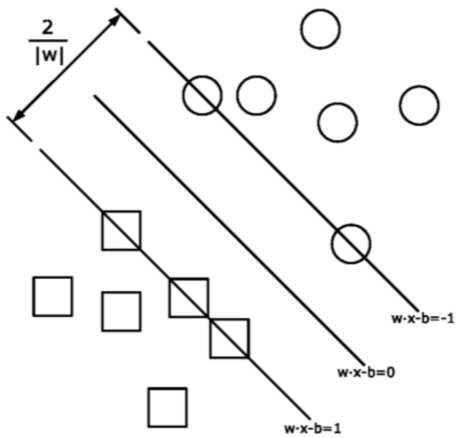

Рис. 3.1. Графічне представлення методу опорних векторів

Алгоритм працює в припущенні, що чим більша різниця або відстань між цими паралельними гіперплощинами, тим меншою буде середня помилка класифікатора. Ми вважаємо, що точки мають вигляд:

$$\{(x_1, c_1), (x_2, c_2), ..., (x_n, c_n)\}$$

де c_i приймає значення 1 або -1, залежно від того, до якого класу належить точка. Кожне x_i - це p-мірний вектор, зазвичай нормалізований значеннями [0,1] або [-1,1]. Якщо точки не будуть нормалізовані, то точка з великими відхиленнями від середніх значень координат точок суттєво вплине на класифікатор. Спочатку розглядається навчальна вибірка, в якій для кожного елемента вже заданий клас, до якого він належить. Необхідно, щоб алгоритм методу опорних векторів класифікував їх таким же чином. Для цього будуємо розділяючу гіперплощину, яка має вигляд:

$$w \cdot x - b = 0.$$

Вектор w - перпендикуляр до розділяючої гіперплощини. Параметр b залежить від найкоротшої відстані гіперплощини до початку координат. Якщо параметр b дорівнює нулю, гіперплощина проходить через початок координат, що обмежує рішення.

Побудова класифікатора

```
clc;
clear all;
load fisheriris
data = [meas(:,1), meas(:,3)];
groups = ismember(species,'setosa');

[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);

svmStruct =
svmtrain(data(train,:),groups(train),'showplot',true);

classes = svmclassify(svmStruct,data(test,:),'showplot',true);
hold on;
plot(6.2, 4.9, 'bo', 'MarkerSize', 10, 'linewidth', 2);
hold on;
plot(5.2, 1.9, 'bo', 'MarkerSize', 10, 'linewidth', 2);

newpoint = [6.2 4.9]
class = svmclassify(svmStruct, newpoint, 'showplot', false)

newpoint = [5.2 1.9]
class = svmclassify(svmStruct, newpoint, 'showplot', false)
```

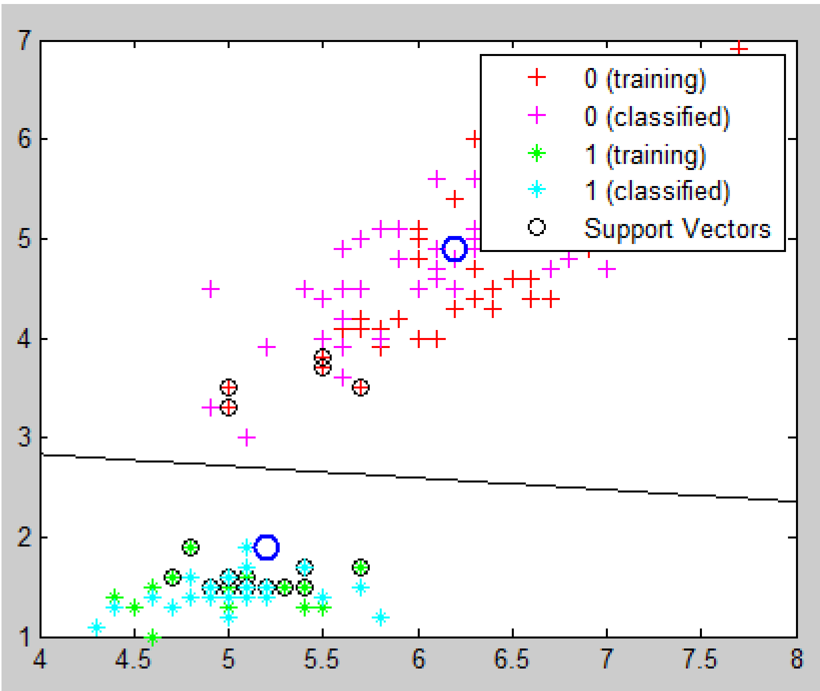

Рис. 3.1. Побудова класифікатора по навчальній вибірці і класифікація нових екземплярів

`load fisheriris` - завантаження вхідних даних в масив meas – в цьому масиві містяться набір даних про іриси (довжина і ширина чашолистка, довжина і ширина пелюстки), масив розмірністю 4x150; і species – в цьому масиві містяться види ірисів (їх назви) з розмірністю 1x150).

`data = [meas(:,1), meas(:,3)];` - створення масиву data з початкового масиву meas. З цього масиву береться перший і третій стовпець з 51 по останній елементи рядка.

`svmtrain(data(train,:),groups(train),'showplot',true);`- функції svmtrain передаються масиви data, group. showplot - параметр відображення графіку з параметром true, що означає що графік буде виведено на екран.

```
class = svmclassify(svmStruct, newpoint, 'showplot', false)
```

класифікує новий екземпляр `newpoint`, використовуючи структуру `svmStruct`, що була створена за допомогою `SVMTRAIN` і повертає імовірний клас `class`

Література

1. Воронцов К. В. Лекции по методу опорных векторов — Режим доступа: http://www.ccas.ru/voron/download/SVM.pdf. — 10.12.2013 г. — Загл. с экрана.
2. Statistics Toolbox™ User's Guide

Контрольні запитання

1. Основна ідея методу опорних векторів.
2. Вбудовані функції Matlab для побудови класифікатора.
3. Вбудовані функції Matlab для визначення класу нового об'єкту.

ПРАКТИЧНЕ ЗАНЯТТЯ №4

РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРОГНОЗУВАННЯ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

Мета роботи – ознайомитися з нейронними мережами на прикладі розв'язання задачі прогнозування міцності керамічних виробів.

Постановка задачі

Незважаючи на великий асортимент, різноманітність форм та фізико-механічних властивостей керамічних виробів, основні етапи їх виробництва (рис.5.1) є загальними і включають такі операції: видобуток сировинних матеріалів, підготовка маси, формування сирцю, сушіння, випалювання, подальша обробка виробів, пакування [1].

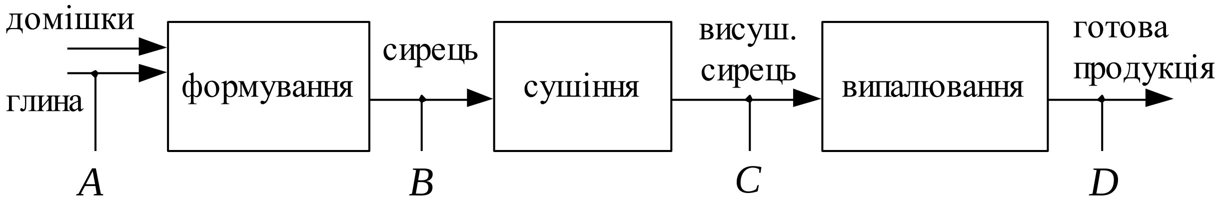

Рис. 5.1. Етапи ТП виготовлення будівельної кераміки

Необхідно: спрогнозувати міцність виробів після закінчення процесу.

Теоретичні відомості

Вектор параметрів сировини $\mathbf{S}$ можна записати як:

$$\mathbf{S} = (\,Al_2O_3,\ CaO,\ MgO).$$

Процес сушіння виробів може відбуватися в тунельній (найбільш використовується в сучасній промисловості) або камерній сушці і характеризується наступними параметрами:

$$\mathbf{D} = (\omega, \tau, W),$$

де τ – час перебування повітря в сушарці;

ω – відносна вологість повітря, яке подається на сушіння;

W – вологість виробів після сушіння.

Процес випалювання на сучасному етапі протікає в тунельних печах, які характеризуються наступними парамтерами:

$$\mathbf{T}_0(p,t) = (T, \Delta t),$$

де T – температурне поле печі;

Δt – інтервал проштовхування виробів.

Таким чином, математична модель прогнозування міцності буде мати вигляд:

$$M = \mathbf{F}(\mathbf{X}),$$

де $\mathbf{X} = (Al_2O_3, CaO, MgO, \omega, \tau, W, \mathbf{T})$ – вектор параметрів ТП.

Побудова нейронної мережі

Моделі на основі нейронних мереж будуються по загальному алгоритму [1], наведеному на рис. 5.1.

Оскільки на адекватність нейромережевої моделі перш за все впливає співвідношення обсягу вхідних даних та параметрів моделі, що виражається співвідношенням:

$$N = W / e,$$

де N – розмір навчальної вибірки; W – число вагових коефіцієнтів; e – точність, моделі, то часто є доцільним зменшення параметрів моделі за рахунок зменшення розмірності вектору вхідних даних .

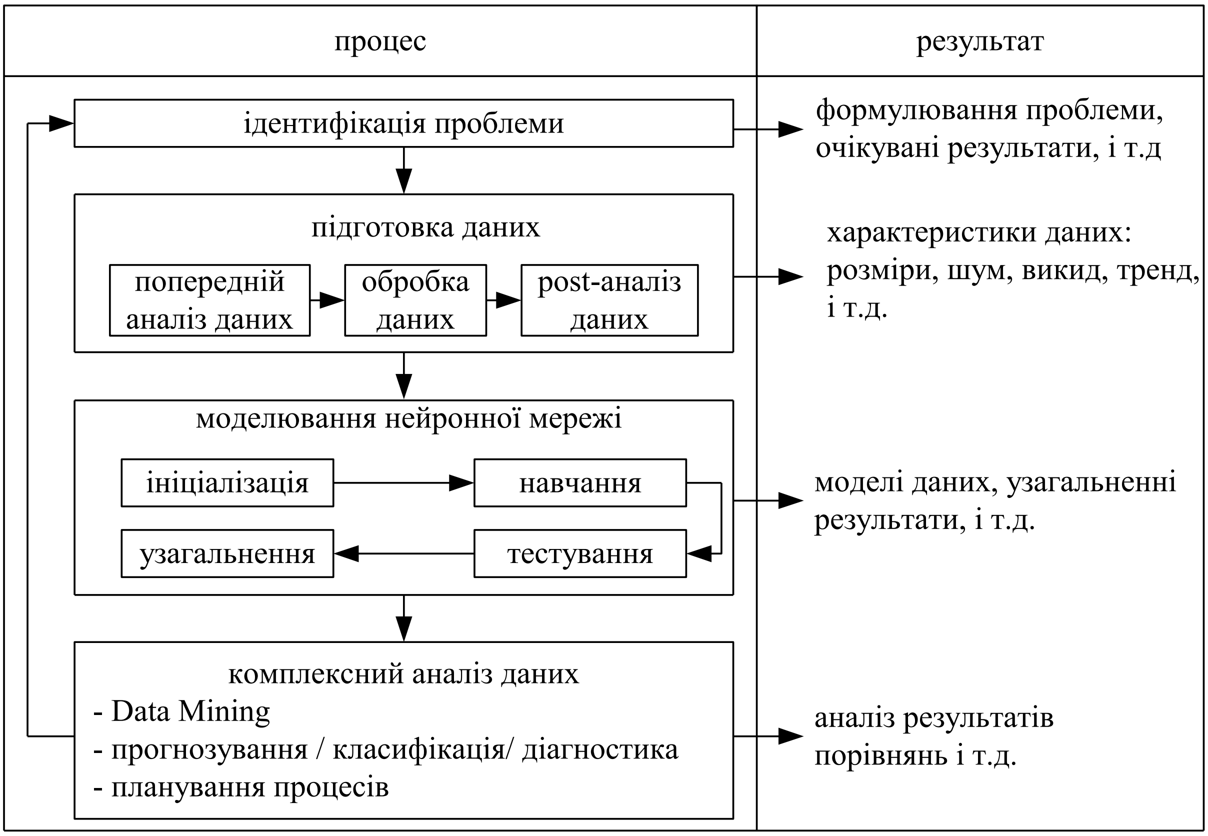

Рис. 5.1. Етапи створення нейромережевої моделі

Програмна реалізація

```
clc;
clear all;

FeatureSize = 21;
TrainSize = 104;
TestSize = 17;

P = dlmread('navch.dat');
for i = 1 : TrainSize
   for j = 1 : FeatureSize
      X(j, i) = P(i, j);
   end
   Y(i) = P(i, FeatureSize + 1);
end
Inp = X; Out = Y;
clear Y X P;

P = dlmread('test.dat');
for i = 1 : TestSize
    for j = 1 : FeatureSize
       Xtest(j, i) = P(i, j);
   end
   Ytest(i) = P(i, FeatureSize + 1);
end
OutTest = Ytest;
clear Ytest;
```

```
% norm input train vector
[normInp, meanInp, stdInp, normOut, meanOut, stdOut] = prestd(Inp, Out);
[InpTrans, transMat] = prepca(normInp, 0.04);
[R,Q] = size(InpTrans)

% norm input test vector
normXtest = trastd(Xtest, meanInp, stdInp);
transXtest = trapca(normXtest, transMat);
normOuttest = trastd(OutTest, meanOut, stdOut);
VV.P = transXtest;
VV.T = normOuttest;

% Create Neural Network
% tansig - activation function of neurons the input and intermediate layers
of Neural Network
% purelin - activation function of neurons the output layers of Neural
Network
% trainlm - the methods of training
net = newff(minmax(InpTrans), [5, 1], {'tansig' 'purelin'}, 'trainlm');

% simulation of Neural Network with train data
[normYsim] = sim(net, InpTrans);
Ysim = poststd(normYsim, meanOut, stdOut);

% calculation of error the training before
E = Out - Ysim;
ErrorBf = mse(E)
clear E;

% training parameters
accuracy = 0.01;
% net.performFcn = 'sse';
net.trainParam.epochs = 200; % number of iteration
net.trainParam.show = 1; % show the training dynamics
net.trainParam.goal = accuracy; % training accuracy

% training neural network
net = train(net, InpTrans, normOut);

% simulation of Neural Network with train data
normYtrain = sim(net, InpTrans);
Ytrain = poststd(normYtrain, meanOut, stdOut);

% calculation of error the training after
E = Out - Ytrain
ErrorAfter = mse(E)
clear E;

% simulation of Neural Network with test data
normYtest = sim(net, transXtest);
Ytest = poststd(normYtest, meanOut, stdOut);

% calculation of error the training after with test data
E = OutTest - Ytest
net.performFcn = 'mse';
ErrorTest = mse(E)
```

```
figure
[m, b, r] = postreg(Ytest, OutTest);
```

В результаті отримаємо

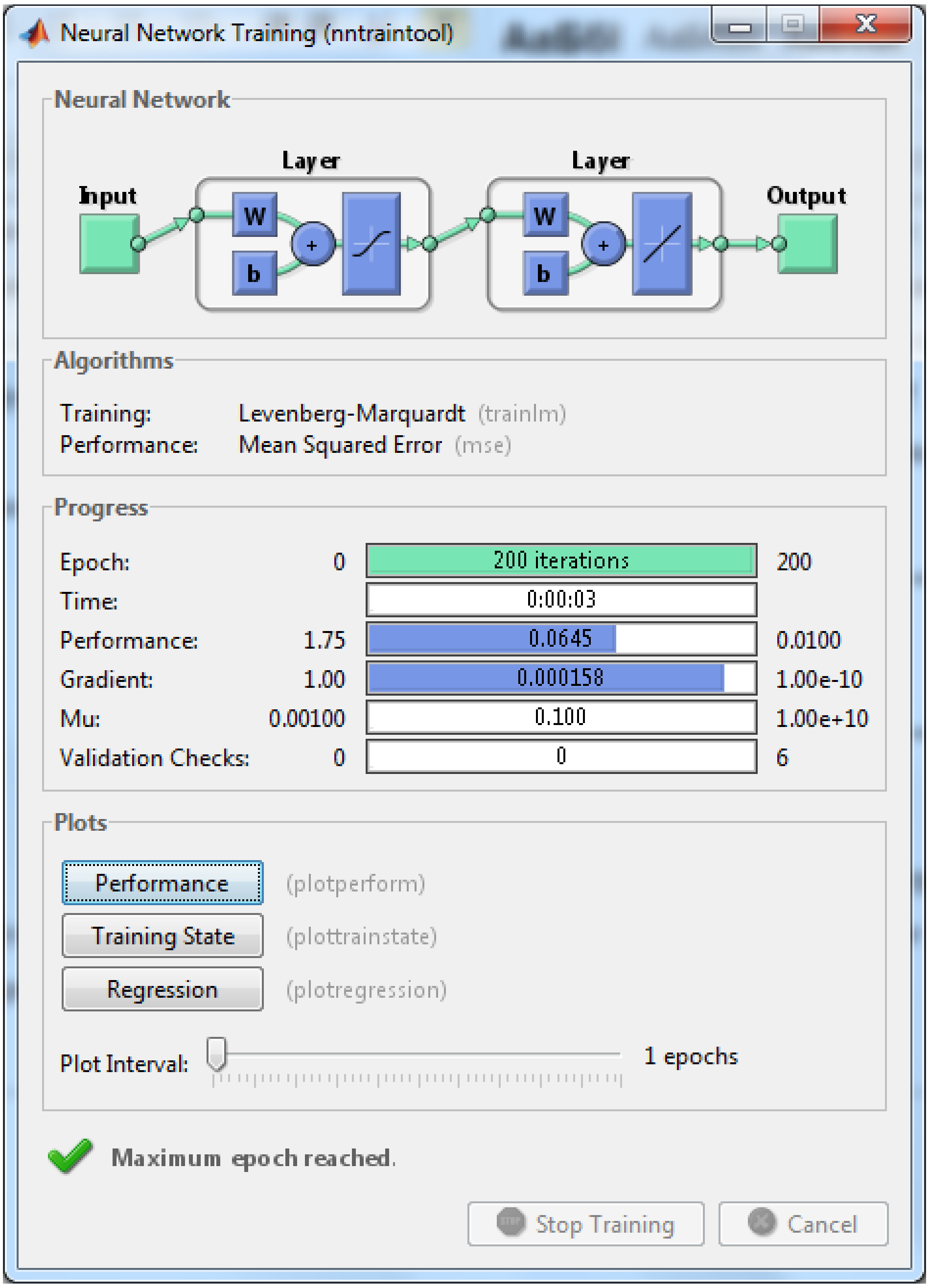

Рис. 5.2. Редактор навчання мережі

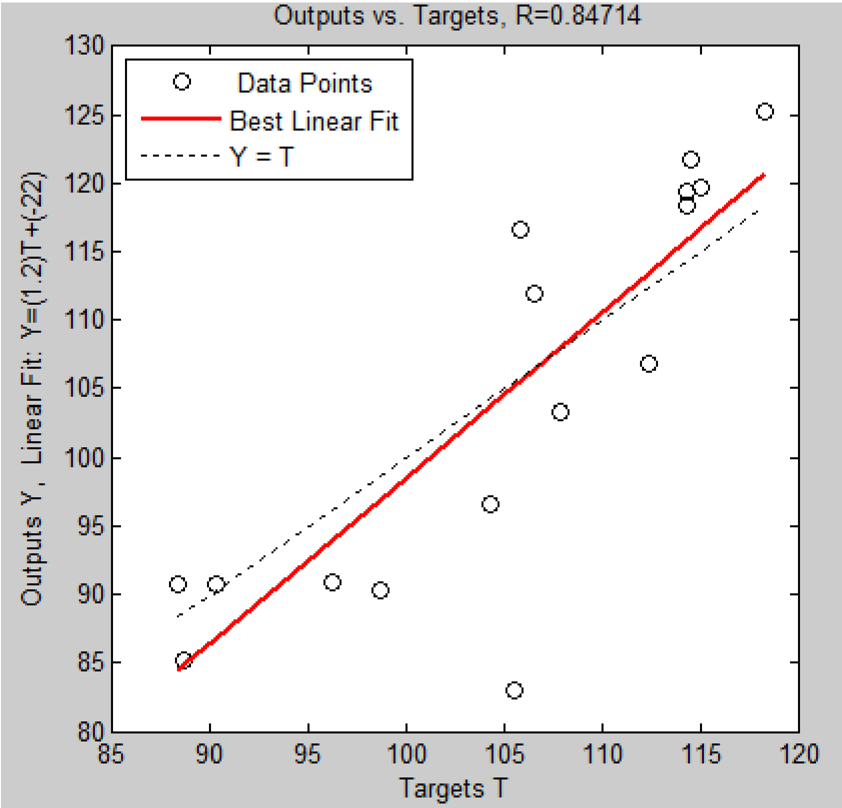

Рис. 5.3. Графік регресії між реальними і спрогнозованими даними

Література

1. Ковалюк Д.О. Моделювання теплотехнологічних об'єктів з розподіленими параметрами: монографія. / Д.О. Ковалюк, С.М. Москвіна – Вінниця: ВНТУ, 2010. – 182 с. – Бібліогр.: с. 163. – 177 пр. – ISBN 978-966-641-337-92.

2. Медведев В.С. Нейронные сети. MATLAB 6 / В.С Медведев, В.Г. Потемкин – М.: ДИАЛОГ-МИФИ, 2002. – 496с– ISBN 5-86404-163-7.

Контрольні запитання

1. Основні етапи побудови нейронної мережі.
2. Функції зменшення розмірності даних в Matlab.
3. Функції побудови нейронних мереж в Matlab.